Knowledge Flows Processes at Multidimensional Intelligent Systems

Konstantin Kostenko

Kuban State University, Krasnodar, Russia

Abstract

The multidimensional intelligent systems formal structure is proposed. It reflects knowledge aspects, developed within multiple subject areas, and al-lows simulating different knowledge representation and processing models created at these areas. The four-dimensional intelligent systems architecture is considered. Unified knowledge representation format of abstract semantic hierarchies' formalisms allows performing formal analysis and achieving acceptable homogeneity for intelligent systems' structural and functional components. Such components adaptation to weakly formalized knowledge models attributes allows creating intelligent systems prototypes that suggest further models' transforming into applied system descriptions. Homomorphisms and homomorphic extensions used for modeling knowledge synthesis processes. Algebraic combinations of knowledge processing operations form basis for developing the goals realization templates at subject domain intelligent system. These templates composed as sequences of diagrams, assigned to intelligent systems structural components. Multidimensional model-ling processes at intelligent systems is implemented as knowledge transfers between such systems' components and knowledge transforming within components. Abstract knowledge processing templates describe the subject areas goals implementation schemes by diagrams, assigned to intelligent systems architecture components. Knowledge processing operations' formal specifications have being applied as such modeling parameters. Ontologies used as foundation for processes' diagrams and templates formal descriptions. This allows creating tools for intelligent systems templates exhaustive processing performed by operations of templates' analysis and control.

Keywords1

Intelligent system, subject area, knowledge representation formalism, cognitive goal, cognitive synthesis, ontology, knowledge processing operation, knowledge-processing diagram

1. Introduction

Mathematical aspects of artificial intelligence concern on investigating and developing the special formal systems. These systems bases on fundamental invariants of abstract mathematics areas. They demonstrate possibility for abstract mathematical entities applications within intelligent system unified and universal components. Such structure realized by variable models adaptation proposed by different knowledge areas that deals with knowledge representation and processing [1-4]. Coordinated knowledge areas attributes representation at intelligent system integrated model allows these attributes joint applying.

Appropriate complete and formalized models creating and exploring is an urgent task of a modern knowledge-based society. The existing diversity of such models develops extensively and is highly heterogeneous. The latter feature is explained by the existence of a tendency to the advantage of specialists' empirical ideas as the basis for artificial intelligence modeling. Intelligent systems general aspects' formal describing within the framework of existing mathematical theories allow us to investigate these aspects by theories' separate entities.

Russian Advances in Artificial Intelligence: selected contributions to the Russian Conference on Artificial intelligence (RCAI 2020), October 10-16, 2020, Moscow, Russia
EMAIL: kostenko@kubsu.ru (K. Kostenko)
ORCID: 0000-0002-9851-2455 (K. Kostenko)

These entities' selected properties are used to model knowledge representation structures and their processing algorithms in accordance with subject area's existing experience. The systematic formalization and theoretical study for knowledge aspects' variety existed at different areas had undertaken within M. Burgin's work [5]. Many significant invariants related to knowledge concept accumulated and formalized in this work. Nevertheless, uniform formal system based on these concepts remains unrealized.

Such systems' initial model constructing should begin with knowledge representation formalisms' abstract invariants. They are homomorphic images for knowledge structures and operations within applied knowledge models. That allows homomorphic extensions' applying for generating the existing knowledge representation models. Investigating the essential elements of abstract knowledge structural representation and structures processing operations is possible for that case. This model's extension based on formalization such knowledge structures and knowledge processing aspects that developed outside mathematics. Aspects' values sets form a multidimensional abstract knowledge space structures. It is convenient for developing such spaces' theory and modeling the intelligent systems' variety founded on application areas peculiarities.

2. Knowledge representation formalisms

The unified and universal approach to abstract knowledge representation implements the class of knowledge representation formalisms definition. Every formalism is a system $\Im = (M, D_M, \circ, \prec)$ with enumerable sets of abstract knowledge M and knowledge fragments D_M, where $M \subseteq D_M$ is solvable in D_M. Special empty knowledge Λ belongs to M and is useful in many ways. Computable composition operation $\circ : D_M \times D_M \to D_M$ applied for knowledge constructing as knowledge fragments combination. Let us suppose that composition operations is injection in cases when its meanings are nonempty. Binary relation on knowledge fragments $\prec \subseteq D_M \times D_M$ is solvable and interpreted as knowledge content inclusion. The knowledge fragments compositions structure describes knowledge algebraic structure. Such structure used as knowledge representation format by knowledge processing algorithms. Such knowledge structure represented by binary tree with leaves marked by elementary knowledge and other vertices – by operation of composition. Relation of content inclusion is basic for creating knowledge properties definitions [7].

Formalism of semantic hierarchies is special knowledge representation formalism. It operates with knowledge representation structures based on binary tree format. Every such formalism defined as pair $H = (M, d)$, where M − enumerable set of abstract knowledge called configurations and d is pair of computable mappings (ε, ψ), with decomposition $\varepsilon : M \to M \times M$ that divide every $z \in M$ onto two parts, and $\psi : M \to R$. Last mapping defines solvable semantic relation satisfied between $\varepsilon(z)$ elements. Set of accepted relations R is enumerable with solvable property of relations' inclusion.

Configuration $z \in M$ called elementary and considered as undividable if $\varepsilon(z) = (\Lambda, \Lambda)$. The configuration structural representation has format of binary tree with elementary configurations assigned to leaves and internal vertices marked by semantic relations that holds for configurations, represented by left and right subtrees of given vertices. The uniform format of knowledge representation formalisms implementing to semantic hierarchies formalisms, applies the set of elementary knowledge extended by set R. Besides that, it is possible to add such specifications for configurations fragments composition operation and con-figurations' inclusion relation that provide keeping properties when transitions implemented from configurations' algebraic structures to binaries trees that rep-resent configurations. This property allows applying formalisms of semantic hierarchies' as universal unified base for abstract knowledge representation at considered intelligent systems architecture [7].

The configurations' tracing relation implements general knowledge inclusion concept for semantic hierarchies' formalisms. It based on existence of binary trees vertices isotonic mappings, where binary trees vertices that connected by mapping have comparable assigned meanings. Class of possible isotonic map-pings splits on subclasses, with mappings of compressing and stretching as important cases. These classes supply intelligent systems, based on knowledge representation formalisms, with special tools for processing knowledge as hierarchical semantic structures.

3. Concept of multidimensional intelligent systems structure

We shall consider multidimensional intelligent systems components architecture, based on independent aspects of knowledge representation and processing, called knowledge dimensions. This allows defining intelligent systems structures by combining from predetermined components. The components general structure depends on knowledge representation separate aspects meanings represented by certain sets. Such properties extracted as result of analysis the transdisciplinary knowledge representations models. They allow constructing separate intelligent systems with components structures selected by examining subject domain properties. Every knowledge dimension settled by finite set of possible knowledge property meanings. They define components properties induced by this dimension. The intuitive dimensions sets serve to be the uniform platform for intelligent systems formal analysis and constructing. The dimension of knowledge abstractness attribute follows to K. Stanovich model [2]. It used three abstractness attribute meanings (quants) that define reflexive, algorithmic and abstract minds for intelligent systems components' classification. These quants distinguish the levels for thinking abstractness representation [2, 8]. Three additional intelligent systems dimensions natural for extending abstractness dimension. They reflect properties of knowledge atomizing, grade and timing. The introduced dimensions and their quantification define intelligent systems cellular structure (see Figure 1).

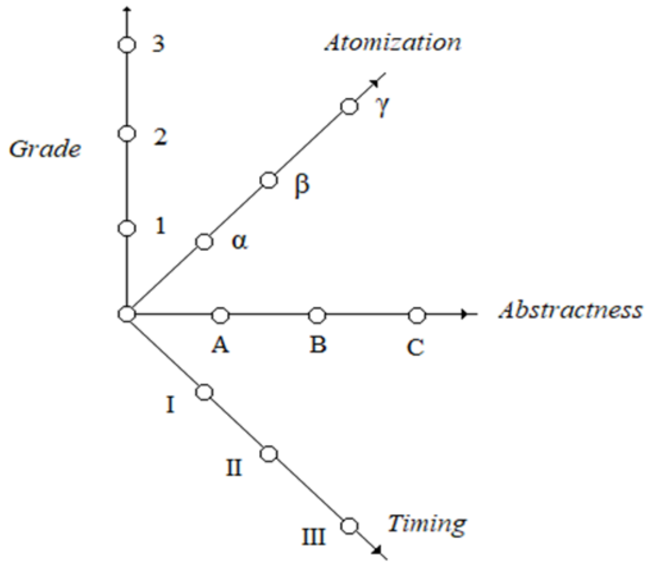

Figure 1: The intelligent systems unified four-dimension cellular architecture

Three added dimensions extend intelligent system components cellular structure. Every new dimension has own quantization by three possible meanings (*full-image*, *partially structured* and *completely atomized* – for knowledge atomization dimension; *superficial*, *acceptable* and *exhaustive* – for knowledge grade dimension; *past*, *present*, *future* – for knowledge timing dimension). The intelligent systems' last two dimensions are new. They extend thinking processes modeling possibilities in comparison with models based on two dimensions [4]. The obtained complete intelligent system structure consists of 81 separate components. Coordinates assigned as quants for intelligent systems separate dimensions identify the structure components. Every component allows addressing by dimensions' quants meanings as four elements sequence $(abstractness, atomization, grade, timing)$. This sequence defines component's unified properties and roles for attached knowledge processing operations with predefined operational and subject domain semantics.

The empty meaning (Λ) for separate dimension measurement describes situation of removal the dimension from intelligent system cellular structure. This allows selecting only necessary system's dimensions when intelligent system with certain properties is constructed. Selected sets of adopted dimensions and dimensions' quants allow introduce intelligent systems classification. It based on such systems properties supported for chosen components sets. In such a way it is possible to define class of expert systems as one-dimensioned intelligent systems based on knowledge atomization dimension with fixed *abstractness* dimension meaning equals to «*algorithmic*» (B). Figure 2 represents the intelligent systems' four-dimension structure that defines class of expert systems. It composed by components of knowledge base, tasks solving processes and professional tasks' solutions (specialists' experience).

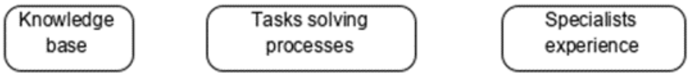

Figure 2: The experts systems unified components structure

Expert systems main components, presented on Figure 2, are based on knowledge atomization dimension with fixed algorithmic level of abstractness. Other dimensions are insignificant for tasks solving processes. Chosen components described by following cellular structure components sequence enumerated from left to right as. $(\alpha, B, \Lambda, \Lambda)$, $(\beta, B, \Lambda, \Lambda)$ and $(\gamma, B, \Lambda, \Lambda)$ or briefly (α, B) , (β, B) and (γ, B) .

The scale of proposed components specifications with simple cells basic structure has deep applied modeling possibilities. Dimensions of abstractness and atomizing allow professional activity modeling at given subject domain with the identical level of specialist's experience. The grade dimension applications allow distinguishing possible meanings for property of knowledge grade and use this dimension at adaptive learning systems with special possibilities for implementing the multilevel education [10]. The timing dimension allows introducing into intelligent systems unified architecture components that model knowledge processing for solving the subject domain problems that are time dependent.

Separate component realizes independent set of knowledge processing flows relevant to applied business processes associated with that component. The component memory represented as complex knowledge structure synthesized by execution the component's business processes. Independent components interactions performed as knowledge transferring and implemented by crossing architecture dimensions with changing quants meanings. Uniform knowledge algebraic structures' format defined as universal for all intelligent system components con-tent representation allows develop uniform collection of abstract operations proposed to be the base for knowledge processing specification.

Applied intelligent system designing technology supposes intelligent systems constructed as several linked universal components of universal cell structure. Simple abstract restrictions on constructing models are concerned on decreasing number of dimensions under consideration and sets of separate dimensions' quants. Cellular structure of intelligent system components allows apply abstract mathematical systems' invariants for such systems properties formal studying. These invariants suggest their simultaneous combined applying as formalizations the invariants provided in cognitive psychology, linguistics and philosophy for thinking processes and knowledge representation. External concepts insertion into mathematical model allows concepts formal aspects specification with further applying for abstract mathematical model extension continue up to applied intelligent systems prototypes.

Every intelligent system components' structure associated with certain knowledge representation formalism. The compliance is necessary for component formal description and external informal and semi-structured knowledge and thinking models invariants. Components' internal life cycles realized independently from other intelligent system model components. The knowledge processing possibility demand components' unified knowledge bases developing.

4. Knowledge transformation and transferring morphisms

Operations of abstract knowledge processing modelled by morphisms for se-mantic hierarchies formalisms. They form morphisms classes' hierarchy based on morphisms types variety used within mathematics different areas. These classes adapted to unified knowledge representation format of semantic hierarchies formalisms. They define abstract ground for modeling the knowledge processing by morphisms combinations. Besides that, the knowledge processing morphisms propose basic classification by ways accepted in mathematics. It includes morphisms of knowledge selecting and transforming morphisms that model function-al invariants for sets and structures transforming, algebraic computations and logic inferences as well as knowledge topological properties based knowledge processing. All these morphisms simulate unary knowledge processing operations with one exception for configuration direct sum (analogous to composition operation at knowledge representation formalism) that is binary.

The uniform set of knowledge classes is demanded for morphisms' domains and ranges. These classes family needs defining in such a way that every morphisms class has certain class' domain and range. Such domains and ranges called morphisms' bases.

Following configurations sets generate well calibrated morphisms bases family as sufficient for intelligent systems and processes versatile mathematic exploration: unstructured knowledge (M), knowledge unstructured fragments (D_M), knowledge of given depth (M_k) with M_0 and M_1 as elementary and simple knowledge subclasses, knowledge semantic hierarchies (Σ), knowledge ordered and unordered series (sets) (S and $\overline{S}$), knowledge neighborhoods of different radiuses and selection predicates described at [9, 11 and 12]. These classes reflect the mathematical models entities adaptations to uniform knowledge representation structures and experience of intelligent systems multidimensional structures designing. Union $M_0 \cup M_1$ has useful interpretation of abstract ontologies. It settled by putting in compliance between elementary knowledge with individuals and simple knowledge with relations between individuals. This implies appearance the new abstract knowledge universal aspects interpreted as based on individuals and relations at subject domains' ontologies. The knowledge composition operation ($\circ$), for knowledge representation formalisms, and configuration direct sum ($\oplus$), for semantic hierarchies' case, useful for creating special knowledge processing operations bases. This allows define operations with several parameters and different parameters domains as unary by parameters' bases compositions (direct sums). Simple convenient format for such composition looks like $(...(B_1 \circ B_2) \circ ...) \circ B_k$ or $(...(B_1 \oplus B_2) \oplus ...) \oplus B_k$ where $B_1, B_2, ..., B_k$ – considered parameters' bases.

Example of morphisms' classes' hierarchy founded on functional invariants within fundamental mathematical models presented below (see Figure 3). Every class has exact formal definition and integrates abstract morphisms adapted to intelligent systems mathematical models.

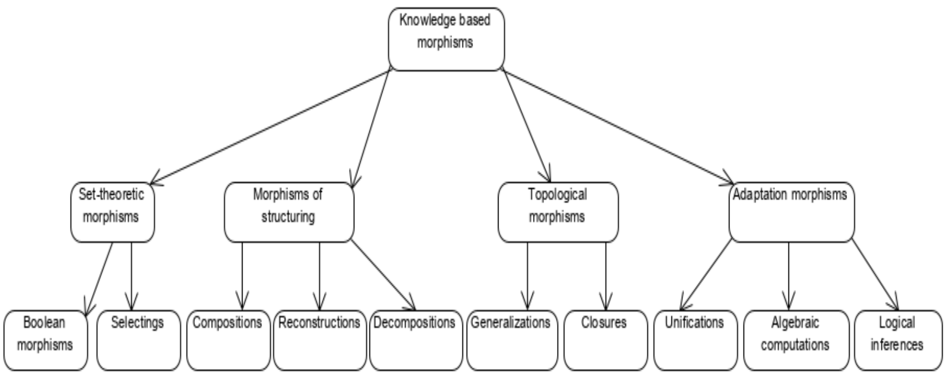

Figure 1: Morphisms hierarchy for knowledge representation formalisms

Detailed knowledge processing morphisms' hierarchy description presented at [9]. Knowledge processing operations offered by given classes realized as morphisms' adaptations to corresponding abstract or applied knowledge domains. Special schemes are necessary for describing the morphisms combining. They describe the complex knowledge processing operations. Following morphisms' properties are important for morphisms adaptation and combining. They represented by concepts of knowledge based homomorphism and homomorphic extension.

Definition. Mapping $h : B_1 \to B_2$, where B_1, B_2 – bases of intelligent systems' mathematical model, is called a homomorphism iff

1) $\forall z_1, z_2 \in D_{M_1} (z_1 \prec z_2 \to h(z_1) \prec h(z_2))$;

2) $\forall z_1, z_2 \in D_{M_1} (h(z_1 \circ z_2) = h(z_1) \circ h(z_2))$.

Symbols $\prec$ and $\circ$ of last definition represent knowledge representation formalisms' inclusion relation and composition operation.

Definition. Mapping $h^{-}: B_1 \to B_2$, where B_1, B_2 – bases of intelligent systems' mathematical model, is called a homomorphic extension iff such homomorphism $h^{+}: B_2 \to B_1$ exists, that $\forall z \in B_1 (h^{+} h^{-}(z) = z)$.

Possibility for modeling the cross-dimensions intelligent system architecture components knowledge transferring without general knowledge properties loosing shows significance of abstract knowledge homomorphisms (homomorphic extensions). Such knowledge transferring are based either on compressing the knowledge structures by appropriate unified knowledge structures tracings or knowledge endomorphisms, as knowledge extensions coordinated by the same tracings [11]. Morphisms types proposed above allow exploring the abstract model morphisms combinations by tools adopted within mathematics. Homomorphic extensions' applied significance defined by possibility of morphisms' consequent specifications up to knowledge processing operation exact descriptions that make models of concrete intelligent systems.

Exact definitions for morphisms classes implemented by formal mathematical expressions. They specify morphisms as mappings with given domains and ranges with predicates satisfied for such mappings. Abstract expressions represent knowledge family referenced to abstract mind level for intelligent systems' abstractness dimension.

Descriptions of knowledge processing scenarios contain references to separate morphisms' classes. Every such class describes stage of goals realization process. Special diagrams present abstract scenarios. They contain the knowledge processing flows functional descriptions. Separate process diagram looks like an oriented graph with vertices marked by morphisms classes. The graph edges describe sequences of morphisms' possible compositions.

This simulates the subject domain tasks solving at intelligent system structural components as realized by sequences of abstract operations performed over abstract knowledge. Such diagrams initiate describing knowledge processing general aspects as first stage of applied intelligent system functionality modeling. Special diagrams describe knowledge processing flows within separate intelligent systems components. Stages of such diagrams gradual developing from abstract ones to detailed knowledge based processes. These stages presented by Figure 4.

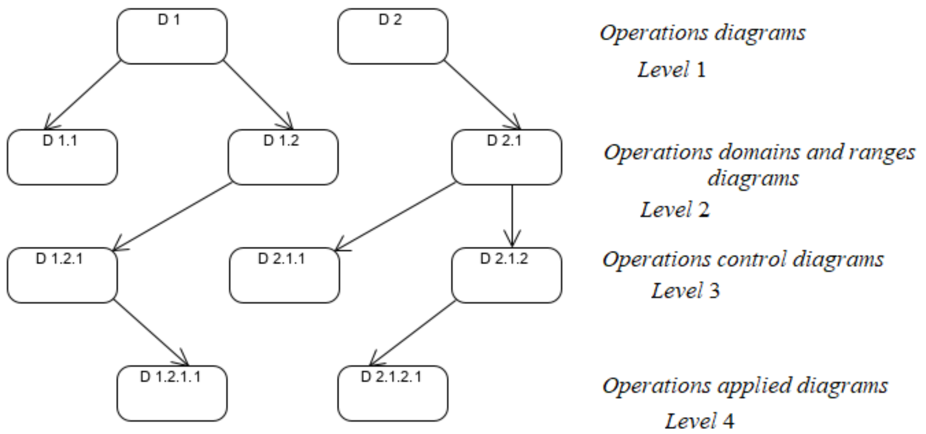

Figure 4: Levels for knowledge processes diagrams descriptions

The first level of established structure accumulates diagrams for subject domain's goals implementation. These diagrams define sequences of activities presented by references on knowledge processing operations classes. Diagrams, placed on presented structure's second level, extended by operations domains and ranges vertices. They present operations inputs and outputs. Domains and ranges may be settled automatically if classes established on first level predetermined as bases for morphisms' classes presented at diagrams. If morphism classes precisely defined, then class vertices for domains and ranges of operations inserted into diagrams automatically. Special cases exist when morphisms' domains and ranges defined as morphisms' bases compositions (direct sums). Other automatically checked condition for second level's diagrams relates to inclusion of diagrams' morphisms' domains and ranges in order of their each after other following. The third level of diagrams' intends for placing describing the diagrams' edges properties. The last ones represented by conditional expressions necessary for control the knowledge flow process through diagram's vertices. The conditions roles supply modeling different schemes of diagrams implementations as morphisms' sequential performing: sequential, parallel, conditional, added with possibility of processes suspension or halting [7].

The forth level contains diagrams represent completely defined applied knowledge flows processes for intelligent systems models. Such diagrams constructed by sequences of transformations initiated by certain diagram on first level. These transformations based on diagrams' changing operations that reduce uncertainness in diagrams' vertices and edges descriptions. Simple form of unique morphism selecting consists in assigning fixed operations to separate diagram's functional vertices given by computing algorithms or algebraic expressions considered as formulas for morphisms' meanings computations. Morphisms' specifications that assigned to diagrams' functional vertices, gradually detailed by adding new elements to morphisms' classes formal descriptions.

The constructed diagrams have hierarchical structure with functional vertices that represented by diagrams are allowed under certain conditions. Additional conditions clarify sub-diagrams' external connections to diagrams vertices linked with functional vertices replaced by sub-diagrams. Several situations possible at that case. Their uniform descriptions may be concerned on properties of content placed into external input and output vertices for given vertex that replaced by diagram. Last condition processed by special procedure that control vertices' replacement correctness. Diagrams used for functional vertices replacing developed separately. Their complete descriptions implemented by sequences of diagrams transformations and presented in multilevel structure of diagrams' constructing processes. If diagram's functional vertex replaced by diagram then input and output vertices of that vertex linked to additional correspondence conditions for inserted diagram's input and output vertices.

The sequence of transforming operations performed one after another generates diagrams' developing chain. Chains' general structure looks like a rooted tree with arbitrary finite sets of child vertices for any internal vertex of the tree Every such tree accumulates different ways of given initial diagram transformations into its specifications related to different diagrams' presentation levels. Diagrams' tree make it possible to integrate specialists' experience of professional tasks solving.

The root vertex of such a tree specifies initial abstract diagram. The tree's other vertices marked by descriptions that define used diagrams' transforming operations. Diagrams' developing processes modelled by chains of diagrams sequentially designed by operations assigned to diagrams' tree vertices. These vertices form ways in the tree with root as processes' starting vertex. The diagrams tree leaves specify unique scenarios of tasks solving within intelligent system component's model. The diagrams defined by trees' leaves and their specifications saved at intelligent systems components' memory. They use applied intelligent system for implementing of actual problems' solving processes at given subject domain.

Every diagram relates to certain component at intelligent system unified structure. Templates for diagrams interconnections describe knowledge flows processes that cross dimensions of intelligent systems unified structure. These flows used for subject domain problems solving, when that knowledge processing performed by operations related to different intelligent systems' components. Templates deal with general models for knowledge flows and realized as cross-dimensions knowledge transferring based on formats for components connections. Every template's vertices linked with components and subject domain tasks solved at these components. It supposed that component' tasks names linked to diagrams of tasks' solving described within components' ontologies. Such ontologies uniform abstract structure considered in this paper consist of four general subareas: component's concepts, properties, tasks and methods. These subareas at component's ontologies allow representing necessary knowledge containing entities and providing these entities applying by component's agents. These agents perform controlling the component's life cycles implementation.

5. Knowledge flows and processes ontology

Processes of templates and diagrams step-by-step constructing for given subject area founded on system of knowledge that reflects systematic view on templates and diagrams realization stages within intelligent system direct development. These stages entities integrated by knowledge maps format define special knowledge flows processes ontology. This ontology relates to entire intelligent system and used for integrated describing the knowledge flows templates. This ontology allows implementing by model based on formalism of semantic hierarchies. It uses sets of elementary and simple knowledge for knowledge flow processes descriptions decompositions. The considered formalism's other knowledge are synthesized by intelligent system agents when knowledge flows modelling performed. Ontology offered and implemented by knowledge map. It contains several separate areas. Every area reflects certain aspect of intelligent systems' knowledge flows processes formalized descriptions. The knowledge flows processes ontology easily constructed by linking appropriate classes at these areas. This ontology compliant with intelligent system unified model entities.

The knowledge maps format belongs to class of semantic networks formats. Maps are convenient for subject domains' ontologies modeling by entities that easily transformed into formats adapted to semantics of descriptive logics. Discussed ontology is made of elementary knowledge entities (individuals). Individuals' properties represented at ontology by special classes with binary relations between classes used as basis for knowledge synthesis. The same names for relations between different pairs of classes allowed in knowledge maps.

Two types of entities possible for elementary knowledge presentation: symbolic names and formal mathematical expressions. Special semantic hierarchies represent names' and expressions' content with demanded accuracy. These hierarchies used as elements of knowledge flow processes. Semantic hierarchies' format used also for knowledge map integrated presenting. The knowledge map special areas serve as knowledge base about the intelligent system dimensions' types and dimensions' quantized meanings. We begin with area for knowledge flow processes ontology area for intelligent systems' goals and components. Possible variant of knowledge map for this area presented below (see Figure 5).

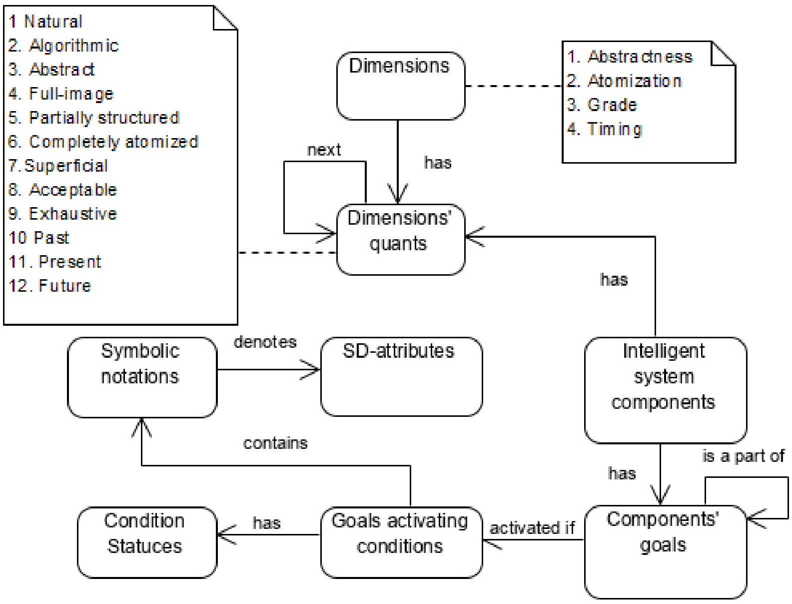

Figure 5: Intelligent system components ontology area

The dimensions' names and dimensions' quants classes defined by dimensions' quants' sets linked there by relation *has*. Dimensions names and their quantized meanings putted into knowledge map as two separate rectangles with the cut-off corner (see Figure 5). The dimension quants' meanings ordered by relation *next*. Intelligent system components' positions uniformly described by components' position into considered four dimensions' components' model. Components' goals class contains names for intelligent system goals functionally correspondent to components. This correspondence modelled by relation has. Goals' class additional ordering realized by *is a part of* relation. The discussed knowledge map area other classes are intended for goals' actuality estimating. These classes' elements used for recognizing goals that need activating and contain conditions of goals activating.

Separate conditions represented as mathematical expressions. They linked with goals and additional parameters introduced for subject domains content measurable attributes. Such attributes meanings reflect situations essential parameters that possible at subject area. The considered knowledge classes form ontology's separate fragment. It includes homogenous classes of completely atomized knowledge representations for knowledge flows processes content based on mathematical expressions and entities' names. This area supports modeling the cognitive operations of knowledge understanding, remembering, applying, ranging, analysis and synthesis [1, 4, and 10].

Templates' and diagrams' ontology main subarea accumulates simple knowledge that allow generate knowledge flows processes' full formal descriptions. Knowledge flows processes algorithmic modeling is similar to thinking processes over mathematical and linguistic content. Let us consider knowledge map example that is possible for such subarea (see Figure 6).

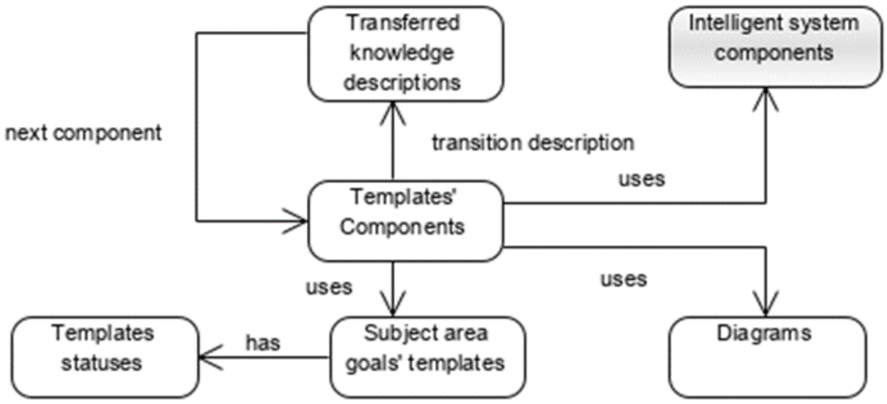

Figure 6: Subject Domain Templates' Ontology Area

Ontology's subarea description presented by Figure 6. It integrates subject domain experience about knowledge flows processes structures and knowledge transfer between components. Considered ontology area contains new class *Subject area goals' templates*. Templates linked externally with subject area goals class. They initiate every template as network composed of Templates' components class' elements. They associated with templates implementing stages at intelligent systems separate components [8]. Every template description defines knowledge flows processes as abstract intelligent system components sequences. Template described as components' sequences includes specifying the knowledge processing diagrams that selected from diagrams' sets assigned to components. Every intelligent system's template component identify unique diagram for knowledge processing within this intelligent system component.

Templates' components sequences, implementing performed by intermediate class of *Transferred knowledge* descriptions. Last class elements are mathematical expressions that describe representation formats for knowledge that transferred between template's linked components. Every such expression specifies transferred knowledge algebraic structure as delivered from *next component* relation first element to the second one. Such structures look as diagrams' operations ranges (domains) compositions. These compositions defined separately and supplied with correspondence that define knowledge transfers between diagrams' operations' domains and ranges.

The considered templates and diagrams ontology last subarea accumulate knowledge about separate diagrams' structures (see Figure 7). The *Diagrams* class within given knowledge map is the same as at Figure 6. Every separate diagram belongs to only one intelligent system's component. It associated with this component by relation *uses*. The diagrams' consecutive transformations represented by *transformed by* relation. Last relation establishes sequences of diagrams' variants (see Figure 4).

This relation serves for developing the diagrams' structures transforming tools. It supply diagrams with developing possibility expressed as diagrams linked by intermediate *Transformations Descriptions* class elements. The considered ontology allows extension by additional classes used for remembering the formal descriptions of diagrams' transforming operations. That imply possibility of mathematical expressions special processing that supply intelligent systems by additional tools for diagrams and templates' control.

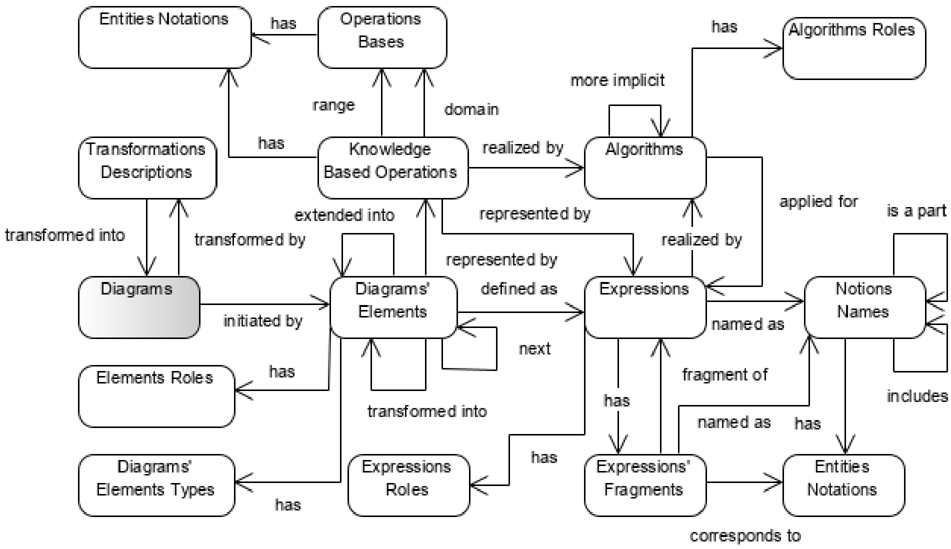

Figure 7: The diagrams knowledge map

The diagrams' names class (*Diagrams*) presents complete diagrams set. The class elements used as roots vertices at diagrams' descriptions and synthesized of their elements' descriptions. Diagrams transformations linked by relations: *extended into* and *transformed into*. Entities of diagrams' structures remembered by their names at class *Diagrams Elements*. Separate diagram's structure described by its elements sequences, connected by next relation (elements following one after another). The relation initiated by used for describing sequences' initial elements attaching to diagrams. The *Diagram's Element Types* (condition, operation or operation base) assigned by relation has. This relation applied for linking *Diagram's Elements* with class *Diagrams' Elements Types* entities. Ontology knowledge map describes diagrams' elements with different type's meanings by correspondent ways. When element is a condition then it described by mathematical formula that represents condition relevant with considered expression role. Such role assigned as condition property and represented by appropriate *Expressions Roles* class entity. Conditional expression supposes own ways for correspondent formulas processing within diagrams' analysis algorithms.

The next examples for conditional expression's roles are possible: *independent*, *disjunctive*, *blocking* or *accumulative*. Conditions roles allow modeling the sequences of performed operations as implementations controlled by attributes. Operations diagrams' elements properties presented by elements' references with Knowledge Processing Operations class entities. The last class contains names for operations. It linked with *Operations Bases* class by relations domain and range with names of bases assigned to appropriate operations. Operations names linking to correspondent mathematical expressions defined by relation *presented by*. These expressions used for describing the operations' formal properties.

The *Algorithms* class initiates computable procedures descriptions intended to different applications' cases. Algorithms can either compute operations and mathematical expressions meanings or analyze their properties. The *Algorithms Roles* class entities represent algorithms' applications ways and cases. If operation has effective algorithm for realization then such operation name also linked with appropriate operation realization algorithm by applying the relation *realized by*. When some algorithm realizes certain operation formal description then it joined with mathematical expression by *applied for* relation.

6. Final conclusions

The intelligent systems' explored model based on components' universal and unified structure. It deals with searching the borders for intelligent systems' effective mathematical foundations. They reflects theories of thinking processes and memory structures offered within different knowledge areas (cognitive science, linguistics, systems engineering and philosophy). These areas invariants' formal descriptions allow applying the mathematics possibilities for creating new technologies of intelligent systems designing.

Two invariants considered as basic for intelligent systems theory. The first one connected with class of knowledge representation formalisms. This invariant's importance is determined by existing formal models' variety that develops extensively and quickly, without serious mathematical exploring the knowledge representation formalism's concept. Formalisms of semantic hierarchies propose unified universal formats for modeling the knowledge representation and processing. They allows integrating and joint applying for knowledge aspects studied at different subject domains. The second proposed invariant relates to intelligent systems' homogenous mathematical structures. The submitted paper's concept of knowledge flows processes was inspired by K. Stanovich idea of knowledge processing levels with one universal dimension at intelligent systems' structures [2]. The two-dimension intelligent systems structure analysis preceded four-dimensional systems' structure developing. It revealed templates and diagrams' importance for knowledge flows processes modeling. This fact explains knowledge flows templates and diagrams significance for the initial stages' of intelligent systems designing technology.

Acknowledgements

The reported study was funded by RFBR, and administration of Krasnodar territory grant project number № 19-41-230008 and by RFBR grant project number № 20-01-00289.

References

[1] Bloom, B.S. Taxonomy of educational objectives: The classification Taxonomy of educational goals. Handbook 1: Cognitive domain / B.S. Bloom, M.D. Engelhart, E.J. Furst, W.H. Hill, D.R. Krathwohl. New York: David McKay, 1956.
[2] Stanovich K.E. Rationality and the reactive mind, Oxford Univ. Press, 2010.
[3] Vaughn Vernon Implementing domain driven design, Addison-Wesley Professional, 2013.
[4] Hermann Helbig: Knowledge Representation and the Semantics of Natural Language, Springer, Berlin, Heidelberg, New York, 2006.
[5] M. Burgin Theory of Knowledge: Structures and Processes. World Scientific Series in Information Studies, Vol. 5, (2017).
[6] Kostenko K.I. Knowledge representation formalisms and intelligent systems structures, Kuban State University, Krasnodar, 2015. (in Russian)
[7] K. Kostenko, A. Lebedeva, B. Levitskii The intelligent office engineering by rational and reactive mind invariants Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, Moscow, Russia, 12-14 November, 2514, 2019, pp.106-116. (http://ceur-ws.org/Vol-2514/paper64.pdf)
[8] Larue, O., Poirier, P., Nkambou, R. A tree-level cognitive architecture for the simulating of human behavior // Canadian AI, LNAI 7310, (2012) 337-342.
[9] Kostenko K. I. Operations of formalized knowledge cognitive synthesis // Programmnaja Ingenerija, 9 (4), (2018), 174-188 (in Russian).
[10] Krathwohl D.R. Revising Bloom's taxonomy: An overview // Theory into Practice. 41 (4), (2002) 212-218.
[11] Kostenko K. I. The rules for abstract knowledge spaces inference operator // Programmnaja Ingenerija, 7(6), (2016) 258 – 267. (in Russian)
[12] Kostenko K. I. Simulation of inference operator for hierarchical knowledge representation formalisms // Programmnaja Ingenerija 7(9), , (2016) 424 – 431. (In Russian).